
Introduction

The Multi-purpose Serial Data Link card facilitates smooth communication between the Meridian 1 system and peripheral devices or D-channels.

Document overview

This document describes the Multi-purpose Serial Data Link (MSDL) card. This card provides multiple interface types with four full-duplex serial I/O ports that can be independently configured for various operations. Peripheral software downloaded to the MSDL controls functionality for each port. Synchronous operation is permitted on all MSDL ports. Beginning with X11 release 19, port 0 can be configured as an asynchronous Serial Data Interface (SDI).

About MSDL

An MSDL card occupies one network card slot in the Meridian 1 Network, CPU/Network, or CE/PE Module and communicates with the CPU over the CPU bus and with I/O equipment over its serial ports. It can coexist with other cards that support the same functions. For example, three cards supported with the MSDL (NT6D80) are QPC757 (DCHI), QPC513 (ESDI), and QPC841 (SDI).

Though the MSDL is designed to coexist with other cards, the number of ports supported by a system equipped with MSDL cards is potentially four times greater than when using other cards. Since each MSDL has four ports, representing a single device, a Meridian 1 can support as many as 16 MSDL cards with a maximum of 64 ports.

MSDL cards are supported on Meridian 1 systems running X11 release 18 and later.

Other documentation

For information on MSDL engineering guidelines, refer to the following documentation:

- *Meridian 1 capacity engineering* (553-3001-149)

For complete discussions of other interface cards, refer to the following documentation:

- *Meridian 1 serial data interface cards description* (553-3001-107)

For information regarding D-channel, Application Module Link, and X11 features, refer to the following documents:

- *ISDN Primary Rate Interface description and administration* (553-2901-100)
- *X11 input/output guide* (553-3001-400)
- *Meridian Link description* (553-3201-110)

To read about the Serial Data Interface (SDI) and Single Terminal Access (STA) features available with the MSDL, see *X11 system management applications* (553-3001-301).

[Figure 1](#) shows connection examples and the number of ports provided by an MSDL, a DCHI, and an ESDI card.

Hardware

The MSDL card is a standard size Meridian 1 circuit card that occupies one network card slot and plugs into the module's backplane connector to interface with the CPU bus and to connect to the module's power supply. On the faceplate, the MSDL provides five connectors, four to connect to I/O operations and one to connect to a monitor device that monitors MSDL functions. [Figure 2](#) illustrates major MSDL components and their locations on the printed circuit card.

Note: Switches S9 and S10 are configured to reflect the device number set in LD 17 (DNUM). S9 designates tens, and S10 designates ones. For example, set device number 14 with S9 at 1 and S10 at 4.

Figure 1
Card connection examples

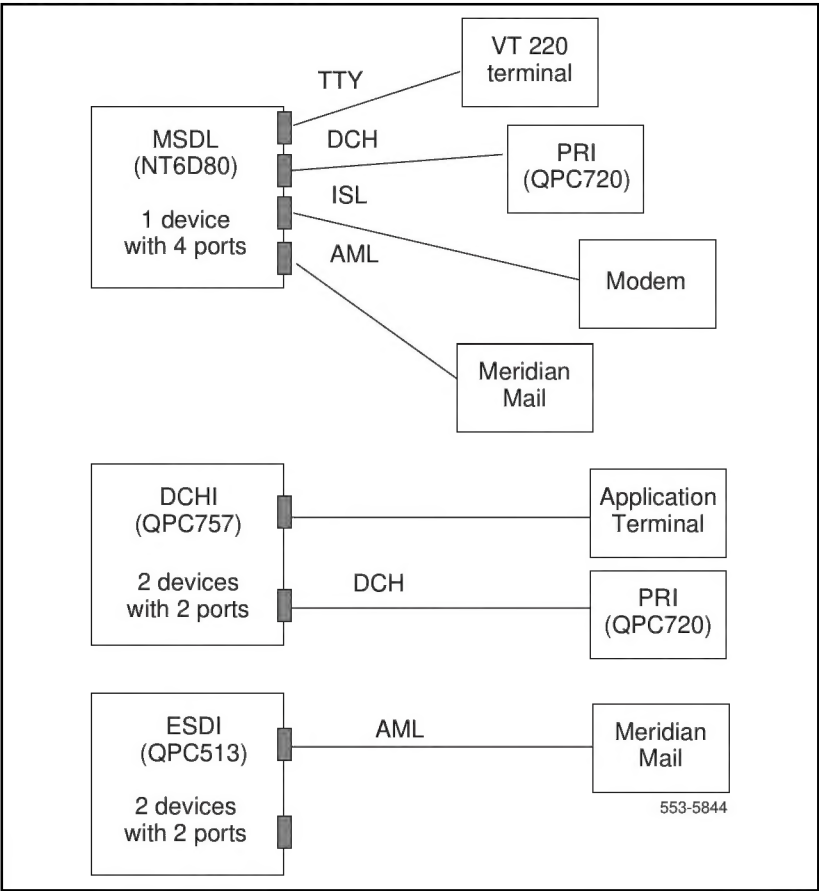
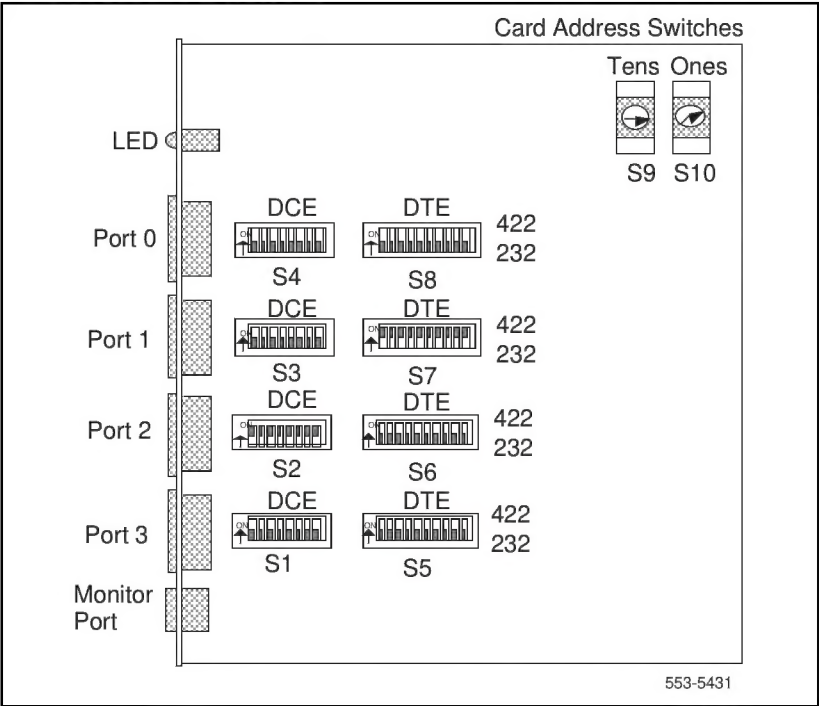


Figure 2
MSDL component layout



Architecture

Two processing units serve as the foundation for the Meridian 1/MSDL operation: the Meridian 1 Central Processing Unit (CPU) and the MSDL Micro Processing Unit (MPU). Meridian 1 software, MSDL firmware, and peripheral software control MSDL parameters. Peripheral software downloaded to the MSDL controls MSDL operations.

The MSDL card's firmware and software do the following:

- communicate with the Meridian 1 CPU to report operation status
- receive downloaded peripheral software and configuration parameters
- coordinate data flow in conjunction with the CPU
- manage data link layer and network layer signaling that controls operations connection and disconnection
- control operation initialization and addressing
- send control messages to the operations

CPU bus interface

The CPU bus transmits packetized information between the Meridian 1 CPU and the MSDL MPU. This interface has a 16-bit data bus, an 18-bit address bus, and interrupt and read/write control lines.

Shared Random Access Memory (RAM) between the Meridian 1 CPU and the MSDL MPU provides an exchange medium. Both the Meridian 1 CPU and the MSDL MPU can access this memory.

Micro Processing Unit (MPU)

The MPU, which is based on a Motorola 68020 processor, coordinates and controls data transfer and port addressing, communicating via the CPU bus with the Meridian 1. Prioritized interrupts tell the MPU which tasks to perform.

Memory

The MSDL card contains two megabytes of Random Access Memory (RAM) for storing downloaded peripheral software that controls MSDL port operations. The MSDL card includes the shared RAM that is used as a communication interface buffer between the CPU and the MPU.

The MSDL Flash Erasable Programmable Read Only Memory (Flash EPROM) also includes the peripheral software to protect it against a power failure or reset. MSDL can copy peripheral software directly from the Flash EPROM after power up or reset instead of requesting that the Meridian 1 CPU download it.

The MSDL card also contains Programmable Read Only Memory (PROM) for firmware that includes the bootstrap code.

Serial interface

The MSDL card provides one monitor port and four programmable serial ports that can be configured for the following various interfaces and combinations of interfaces:

- synchronous port 0–3
- asynchronous port 0 (beginning with X11 release 19)
- DCE or DTE equipment emulation mode
- RS-232 or RS-422 interface

Transmission mode All four ports of the MSDL can be configured for synchronous data transmission by software. Port 0 can be configured for asynchronous data transmission for CRT, TTY, and printer applications only.

Equipment emulation mode Configure an MSDL port to emulate DCE or DTE by setting switches on the card and downloading LD 17 interface parameters.

I/O port electrical interface Each MSDL port can be configured as an RS-232 or RS-422 interface by setting the switches on the MSDL card. MSDL ports use Small Computer Systems Interface (SCSI) II 26-pin female connectors.

Figure 3 illustrates the MSDL functional block diagram. The MSDL card is divided into four major functional blocks:

- CPU bus interface
- Micro Processing Unit (MPU)
- Memory
- Serial interface

Figure 4 shows Meridian 1 architecture using the MSDL as an operational platform. It illustrates operation routing from the Meridian 1 CPU, through the MSDL, to the I/O equipment. It also shows an example in which DCH operation peripheral software in the MSDL controls functions on ports 2 and 3.

MSDL operations

The Meridian 1 automatically performs self-test and data flow activities. Unless a permanent problem exists and the system cannot recover, there is no visual indication that these operations are taking place.

The Meridian 1 controls the MSDL card with software that it has downloaded. The MSDL and the Meridian 1 enable the MSDL by following these steps:

- 1** When the MSDL card is placed in the Meridian 1, the card starts a self-test.
- 2** When the MSDL passes the test, it indicates its state and L/W version to the Meridian 1. The Meridian 1 CPU checks to see if downloading is required.
- 3** After downloading the peripheral software, the Meridian 1 enables the MSDL.
- 4** MSDL applications (DCH, AML, SDI) may be brought up if appropriately configured.

Figure 3
MSDL block diagram

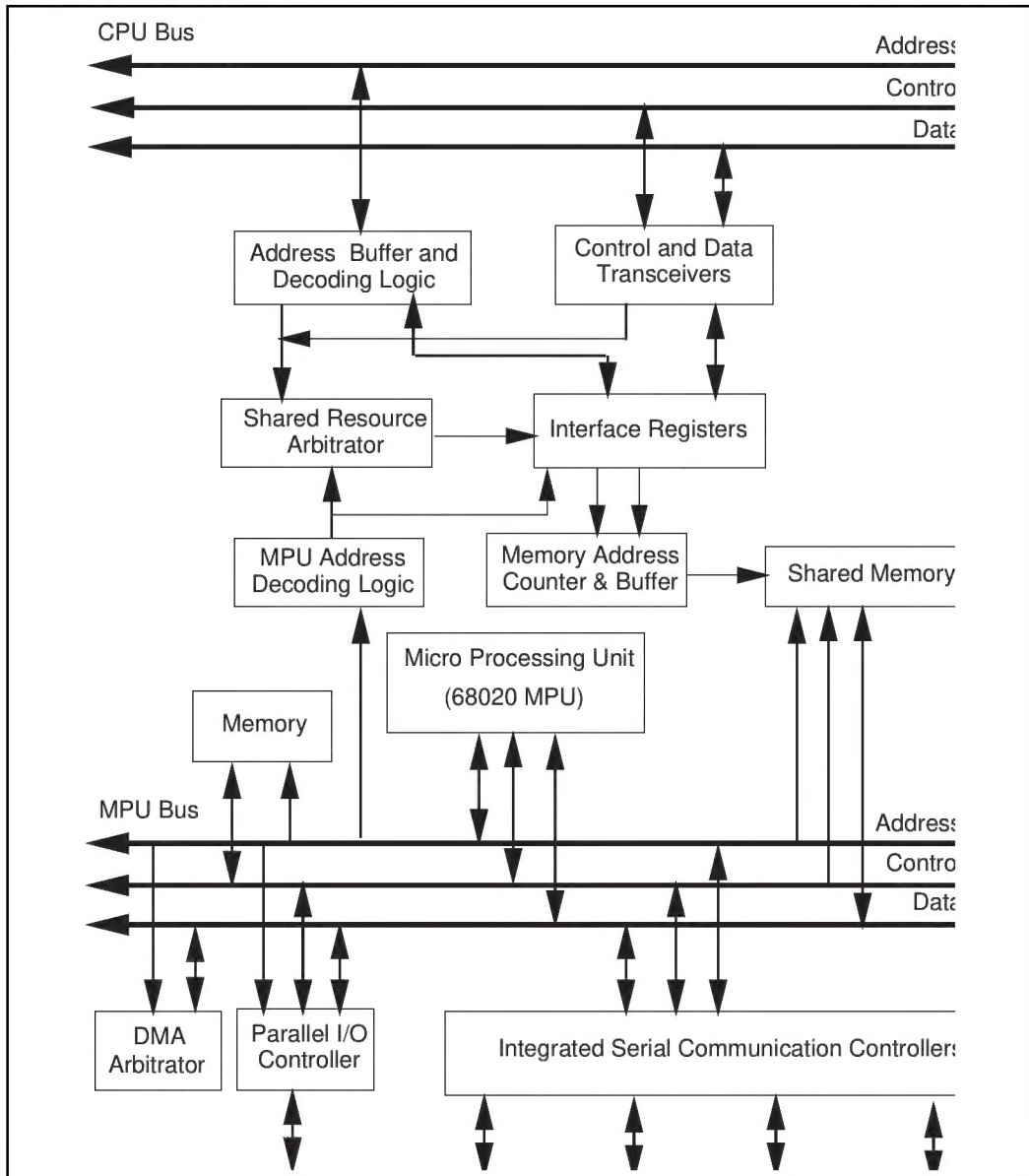
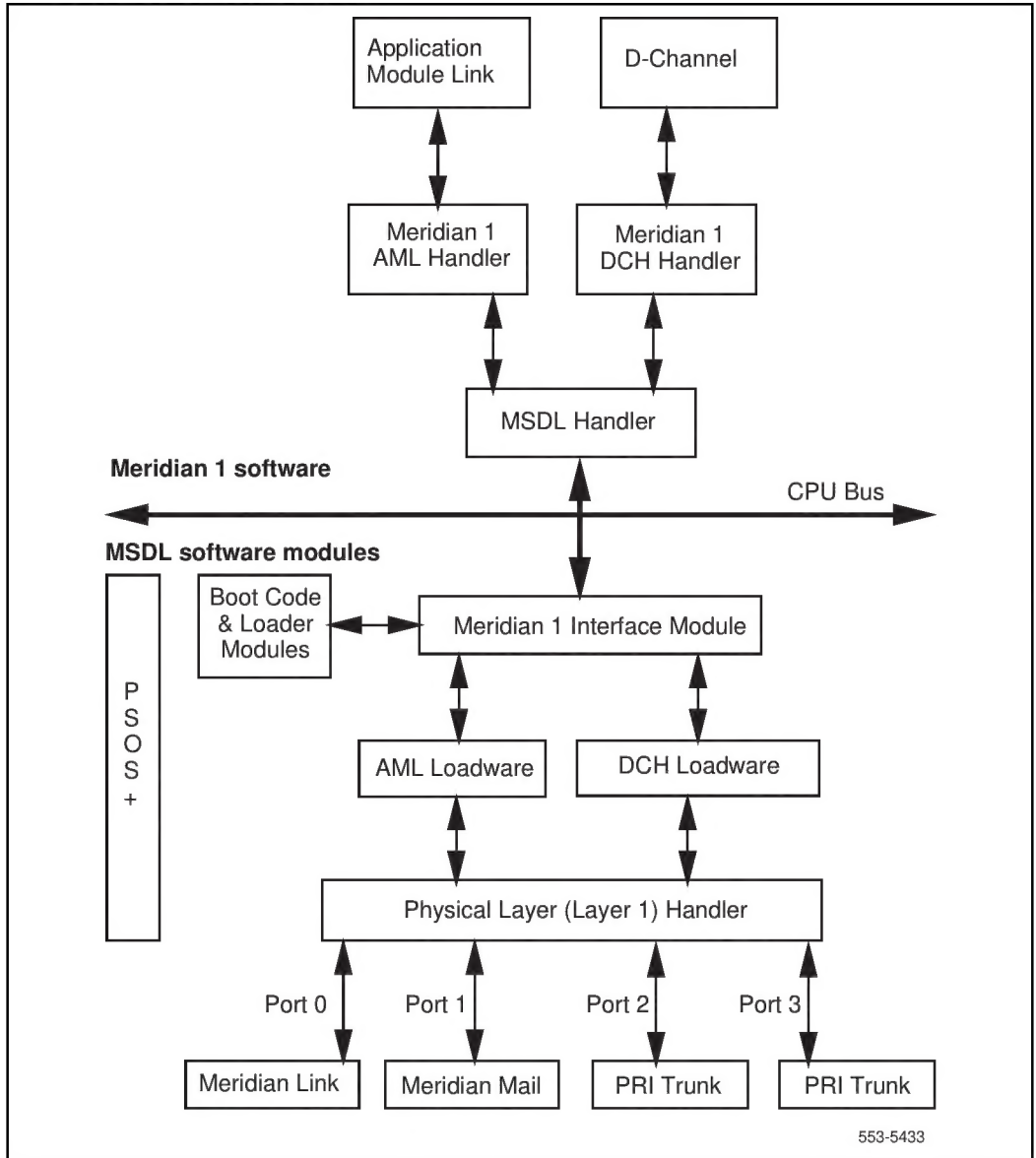


Figure 4
MSDL functional block diagram



Data flow

The MSDL transmit interface, managed by the MSDL handler, sends data from the Meridian 1 to the MSDL. This interface receives packetized data from the Meridian 1 and stores it in the transmit buffer on the MSDL. The transmit buffer transports these messages to the appropriate buffers, from which the messages travel over the MSDL port to the I/O equipment.

The MSDL uses the MSDL receive interface to communicate with the Meridian 1. The MSDL card receives packetized data from the I/O equipment over the MSDL ports. This data is processed by the MSDL handler and sent to the appropriate Meridian 1 function.

The flow control mechanism provides an orderly exchange of transmit and receive messages for each operation. Each operation has a number of outstanding messages stored in buffers waiting to be sent to their destinations. As long as the number of messages does not exceed the threshold specified, the messages queue in the buffer in a first-in-first-out process.

If the outstanding number of messages for an operation reaches the threshold, the flow control mechanism informs the sender to wait until the number of messages is below the threshold before sending the next message.

If buffer space is not available, the request to send a message to the buffer is rejected and a NO BUFFER fault indication is sent.